Distributed Solution of the Blendshape Rig Inversion Problem

Stevo Racković
stevo.rackovic@tecnico.ulisboa.pt
Instituto Superior Técnico
Lisbon, Portugal

Cláudia Soares
claudia.soares@fct.unl.pt
NOVA School of Science and Technology
Caparica, Portugal

Dušan Jakovetić
dusan.jakovetic@dmi.uns.ac.rs
University of Novi Sad
Novi Sad, Serbia

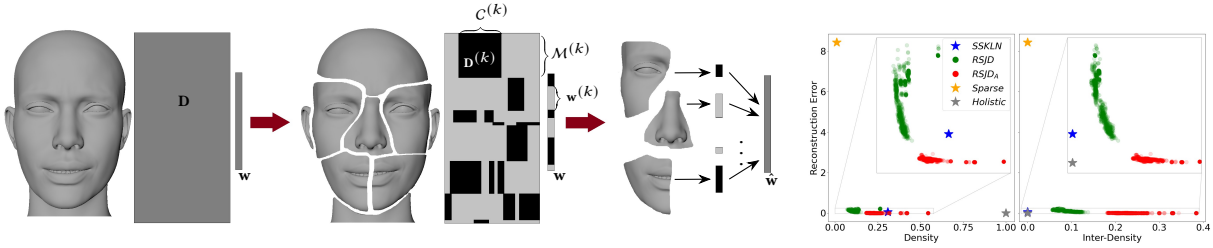

Figure 1: Left: A blendshape offset matrix D is clustered in both mesh (rows) and controller (columns) space, so the face model is divided into several submodels. The inverse rig problem is solved for each cluster and the results are aggregated into the prediction $\hat{\mathbf{w}}$. Face model ©MetaHuman Creator. Right: Trade-off of the reconstruction error (E_R) versus the density (E_D) and versus the inter-density (E_{ID}), of the clustered blendshape matrix, for different clusterings. Each dot represents a single clustering output, with $4 \leq K \leq m = 102$, for *RSJD* and $RSJD_A$.

ABSTRACT

The problem of rig inversion is central in facial animation, but with the increasing complexity of modern blendshape models, execution times increase beyond practically feasible solutions. A possible approach towards a faster solution is clustering, which exploits the spacial nature of the face, leading to a distributed method. In this paper, we go a step further, involving cluster coupling to get more confident estimates of the overlapping components. Our algorithm applies the Alternating Direction Method of Multipliers, sharing the overlapping weights between the subproblems and show a clear advantage over the naive clustered approach. The method applies to an arbitrary clustering of the face. We also introduce a novel method for choosing the number of clusters in a data-free manner, resulting in a sparse clustering graph without losing essential information. Finally, we give a new variant of a data-free clustering algorithm that produces good scores with respect to the mentioned strategy for choosing the optimal clustering.

CCS CONCEPTS

• **Computing methodologies** → **Motion processing**; **Distributed algorithms**; Cross-validation.

KEYWORDS

blendshape animation, face segmentation, inverse rig problem

SA Technical Communications '23, December 12–15, 2023, Sydney, NSW, Australia

ACM ISBN 979-8-4007-0314-0/23/12.
https://doi.org/10.1145/3610543.3626166

ACM Reference Format:
Stevo Racković, Cláudia Soares, and Dušan Jakovetić. 2023. Distributed Solution of the Blendshape Rig Inversion Problem. In *SIGGRAPH Asia 2023 Technical Communications (SA Technical Communications '23), December 12–15, 2023, Sydney, NSW, Australia.* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3610543.3626166

1 INTRODUCTION

Blendshape animation deforms a 3D mesh $\mathbf{b}_0 \in \mathbb{R}^{3n}$ by interpolating between a set of blendshapes $\mathbf{b}_1, ..., \mathbf{b}_m \in \mathbb{R}^{3n}$, where n is the number of mesh vertices [Lewis et al. 2014]. Blendshapes represent different shapes the mesh can take, and blending them as a weighted sum generates wide range of shapes $f(\mathbf{w}; \mathbf{B}) = \mathbf{b}_0 + \mathbf{B}\mathbf{w}$, where the weights $\mathbf{w} = [w_1, ..., w_m]$ define the amount of influence each blendshapes provides, and $\mathbf{B} \in \mathbb{R}^{3n \times m}$ is a blendshape matrix created by stacking the blendshape vectors as its columns. In modern blendshape models, additional corrective terms might be included in the rig function [Racković et al. 2023a].

A problem considered in this paper is the inversion of the rig. I.e., given a target mesh $\widehat{\mathbf{b}} \in \mathbb{R}^{3n}$, find a configuration of the weighs $\mathbf{w}$ that closely approximates the target. Model-based solutions of the rig inversion exploit the structure of the rig functions and rely on optimization techniques rather than data [Çetinaslan 2016; Racković et al. 2023b]. A SOTA model-based solution is given in [Racković et al. 2023a], and it solves the problem

$$\underset{\mathbf{0} \leq \mathbf{w} \leq \mathbf{1}}{\text{minimize}} \ \frac{1}{2}\|f(\mathbf{w}) - \widehat{\mathbf{b}}\|_2^2 + \alpha \mathbf{1}^T \mathbf{w}, \tag{1}$$

using coordinate descent. We will rely on this approach in our experiments, although the proposed method can work with an arbitrary model-based inverse rig solver.

The human face has local nature, hence most of the vertices are irrelevant for estimating the majority of the weights. This calls

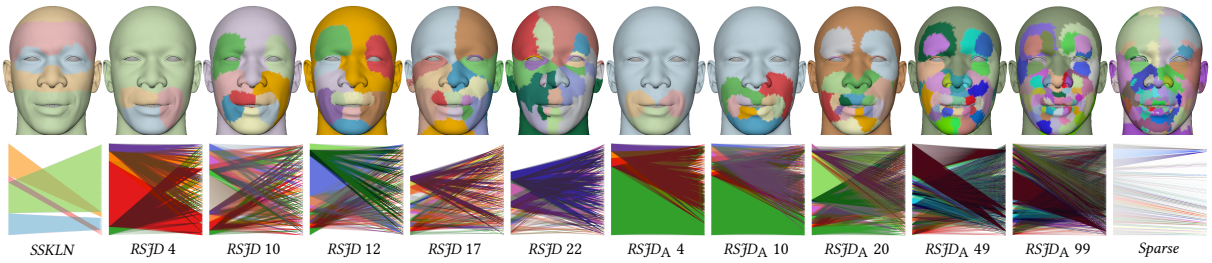

Figure 2: Clustering outputs of the four approaches — for the *RSJD* and *RSJD*$_A$, the number of clusters K is indicated. Besides the mesh clusters, the figure shows bipartite graphs consisting of the vertices (left) and controllers (right). Each color indicates a single cluster, with edges representing the cluster correspondences. (The avatar *Jesse* ©MetaHuman Creator.)

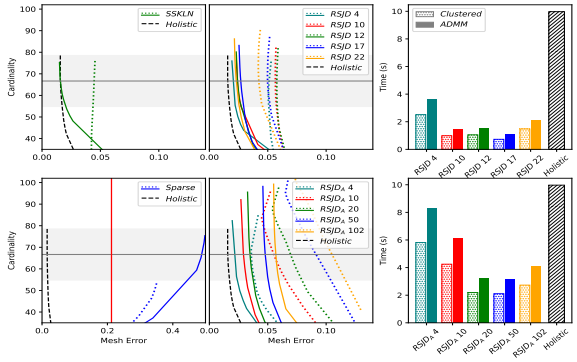

Figure 3: Left: Trade-off of the mesh error (RMSE) vs cardinality, as functions of the regularization parameter $\alpha > 0$. The dotted lines represent a naive clustered solution, and solid the ADMM solution. A dashed line shows a holistic approach. A horizontal line shows the cardinality of the ground-truth data, with a shaded standard deviation. A red line represents a baseline where w = 0. Right: The average execution time.

for a segmented model, where objective (1) is split into subproblems with only relevant weights estimated over each mesh segment (Fig. 1). Early works split the face manually [Choe and Ko 2006]. Some models propose segmenting the mesh based on the vertex behavior over animated sequences [Tena et al. 2011], which makes the clusters susceptible to the quality of the data, and unsuitable for model-based approaches. In [Seol et al. 2011], the mesh regions are painted manually and blendshapes are assigned to the corresponding segments. In [Racković et al. 2021], mesh clusters are estimated from a given blendshape matrix. While clustering helps reduce the size of problem (1), and regularize the solution, question is what to do with the weights shared between multiple clusters. Such values can be averaged, but if the coupling between the clusters is included in the optimization, the shared estimates improve, as we will show.

Contributions. I) We formulate a metric to evaluate the goodness of the blendshape clusters, based on the sparsity and the quality of reconstruction of a given clustering, apriori to the fitting phase, yielding the optimal number of clusters K in a data-free manner.
II) We propose an adjustment to the blendshape assignment within the clustering technique of [Racković et al. 2021], which, results in a denser graph but a higher reconstruction quality.
III) We propose a model-based solution to the inverse rig problem in a clustered setting. The proposed method applies the alternating direction method of multipliers (ADMM) [Boyd et al. 2011], in combination with coordinate descent similar to [Racković et al. 2023a], allowing coordination between the clusters.

2 METHOD

The above contributions build into the following pipeline. Several instantiations of the clusterings are made and evaluated based on the proposed metric for estimating reconstruction error vs density trade-off, in order to choose the best representative clustering. This pipeline can work with an arbitrary clustering method, as shown here. The clusters are used to solve the inverse rig in a distributed manner, where ADMM allows the coupling of the overlapping components, as opposed to a naive clustered solution. The pipeline produces solutions closely matching that of the holistic approach in terms of sparsity and accuracy, while significantly reducing the execution time (50% reduction). A naive clustered solution demands slightly less time than a proposed method, but does not compare with our solution in accuracy or the sparsity metric. Supplementary videos show a clear superiority of our results.

2.1 Clustering of the Face

The clustering methods of [Seol et al. 2011] (here termed *SSKLN*, from the authors' initials) and of [Racković et al. 2021] (termed *RSJD*) transform the blendshape matrix $\mathbf{B} \in \mathbb{R}^{3n\times m}$ into a matrix of offset values $\mathbf{D} \in \mathbb{R}^{n\times m}$. *SSKLN* assumes that an artist manually selects the four mesh segments, and each blendshape i is assigned to a relevant mesh segment if a magnitude of deformation over the segment $\mathcal{M}^{(k)}$ is at least half of its magnitude of deformation over the entire mesh, producing in this way $K = 4$ controller clusters $C^{(k)}$. *RSJD* performs K-Means clustering over the rows of $\mathbf{D}$, to obtain mesh clusters $\mathcal{M}^{(k)}$, for $k = 1, ..., K$. Then, K-means is applied over each blendshape vector to split it in two subvectors, one with high entries and the other one with low. The controller i is assigned to clusters corresponding to the high-valued labels. K is slected by user based on cross-validation.

Proposed Clustering Method. Assigning blendshapes to the mesh segments where their effect is significantly larger than in the others does not imply that their effect within the corresponding cluster

will be significant compared to other blendshapes. We propose an adjustment to *RSJD* wrt this — the lowest magnitude value among the blendshapes assigned to an observed cluster is taken as a threshold, $p^{(k)} = \min \sum_{i \in \mathcal{M}^{(k)}} (b_j^i)^2$ for $j \in C^{(k)}$. Consequently, all the blendshapes such that $\sum_{i \in \mathcal{M}^{(k)}} (b_l^i)^2 > p^{(k)}$ for $l \notin C^{(k)}$, are assigned to the cluster as well. This method will be termed $RSJD_{\text{A}}$ ("*A*" standing for "*adjusted*").

Choosing the Number of Clusters K. Consider a blendshape matrix $\mathbf{B} \in \mathbb{R}^{3n \times m}$, segmented into submatrices $\mathbf{B}^{(k)} \in \mathbb{R}^{3n_k \times m_k}$, for $k = 1, ..., K$, (Fig. 1). The *Density* of the clustering represents the percentage of the kept elements of the matrix, $E_D = \sum_{k=1}^{K} \frac{n_k m_k}{nm}$, where $n_k = |\mathcal{M}^{(k)}| < m$ and $m_k = |C^{(k)}| < m$ are the number of vertices and the number of blendshapes assigned to cluster k. We can understand this as a number of edges E in a bipartite graph $G = (U, V, E)$ where U represents the vertices of the mesh and V the controllers — an edge $(i, j) \in E$ is drawn for every $i \in \mathcal{M}^{(k)}$ and $j \in C^{(k)}$, for $k = 1, ..., K$ (Fig. 2). *Inter-Density*, E_{ID}, gives the size of the clusters' overlap — the number of edges shared between multiple clusters in the bipartite graph G, that is, edges $(i, j) \in E$ such that $j \in C^{(k_1)} \cap C^{(k_2)}$ for $k_1, k_2 = 1, ..., K$, and $k_1 \neq k_2$. It indicates how much coupling should be added between the clusters in the fitting phase. For measuring the *Reconstruction Error*, we focus on the ratio between the kept and dismissed elements. Observe the submatrices $\bar{\mathbf{B}}^{(k)} \in \mathbb{R}^{3n_k \times (m - m_k)}$, for $k = 1, ..., K$, which represent rejected elements of $\mathbf{B}$. Compute the sum of the squared entries of all these matrices $E_{R1} = \sum_{k=1}^{K} \sum_{i=1}^{n_k} \sum_{j=1}^{m-m_k} (\bar{B}_{ij}^{(k)})^2$, and a sum of the kept elements as $E_{R2} = \sum_{k=1}^{K} \sum_{i=1}^{n_k} \sum_{j=1}^{m} (B_{ij}^{(k)})^2$. The reconstruction error is now $E_R = E_{R1}/E_{R2}$.

An optimal clustering should exhibit small E_D and E_R. The holistic case has $E_R = 0$ and $E_D = 1$. Fig. 1 shows the results of clustering methods — *SSKLN* is deterministic, while the other two can vary with K; hence, we repeat the clustering 1000 times, for $4 \leq K \leq m = 102$. We also consider the extremely sparse case, where each mesh vertex is assigned to exactly one blendshape that produces the largest offset (termed *Sparse*). The instances of *RSJD* are closer to the lower left corner (left subfigure) than *SSKLN* or $RSJD_{\text{A}}$. $RSJD_{\text{A}}$ leads to low E_R, but E_D can get large, while *RSJD* is of lower density but higher E_R. In the case of E_R versus E_{ID}, the distinction between *RSJD* and $RSJD_{\text{A}}$ is even cleaner, however, *SSKLN* has $E_{ID} = 0$ as its clusters do not overlap. An optimal choice of the clustering should be based on these plots, choosing the point near the elbow of the trade-off curve. For $RSJD_{\text{A}}$, this would be one of the clusterings with low E_D, and for *RSJD*, the one with low E_R. We proceed to work on several choices of K and we show that a standard cross-validation leads to the same conclusions on the choice of K, validating that the considered K-selection works. In Fig. 2 (and Table 1 of the supplement), we show obtained clusters.

2.2 Distributed Solution to the Rig Inversion

The objective function for the inverse rig problem, as formulated in [Racković et al. 2023a], is (1). In the clustered setting, one can split this into subproblems

$$\underset{\mathbf{0} \leq \mathbf{w}^{(k)} \leq \mathbf{1}}{\text{minimize}} \frac{1}{2} \| f^{(k)}(\mathbf{w}^{(k)}) - \widehat{\mathbf{b}}^{(k)} \|_2^2 + \alpha^{(k)} \mathbf{1}^T \mathbf{w}^{(k)}, \tag{2}$$

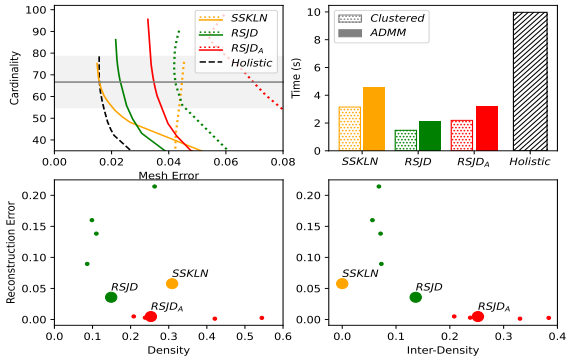

Figure 4: Top left: Trade-off between the cardinality and RMSE, as functions of $\alpha > 0$. Top right: The average execution time per frame. Bottom: Trade-off between E_R and E_D and between E_R and E_{ID}. Large dots indicate a chosen clustering, while the smaller ones represent discarded cases.

for $k = 1, ..., K$, where $\mathbf{w}^{(k)} \in \mathbb{R}^{m_k}$ is a vector containing only the m_k weights assigned to the cluster k; $\widehat{\mathbf{b}}^{(k)} \in \mathbb{R}^{3n_k}$ is a subvector of the target mesh $\widehat{\mathbf{b}}$, consisting of the n_k vertices from the corresponding cluster, and $f^{(k)}(\cdot)$ is a blendshape function restricted only to the vertices and controllers within the cluster k (Fig. 1). If these subproblems are solved independently, they yield a set of local weight vectors $\hat{\mathbf{w}}^{(k)}$, that should be merged into a single global prediction vector $\hat{\mathbf{w}}$. For the controllers that are shared among multiple clusters, the final value is taken as the average of all the estimates. More formally, we introduce the mapping from local variable indices into a global variable index as $j = \mathcal{G}(k, i)$, i.e., for some local variable $\mathbf{v}^{(k)}$ and a global variable $\mathbf{v}$, a local component $(\mathbf{v}^{(k)})_i$ corresponds to the global component $\mathbf{v}_j$. A diagonal matrix $\mathbf{S} \in \mathbb{R}^{m \times m}$ has entries corresponding to the multiplicity of each controller over the clusters, i.e., $S_{ii} = \sum_{k=1}^{K} \sum_{\mathcal{G}(k,i)} 1$. Now, the global weight estimate is obtained as $\hat{\mathbf{w}} = \mathbf{S}^{-1} \sum_{k=1}^{K} \mathbf{v}^{(k)}$, where the entries of $\mathbf{v}^{(k)} \in \mathbb{R}^m$ are the values of $\hat{\mathbf{w}}^{(k)}$ obtained for the corresponding cluster, i.e., $(\mathbf{v}^{(k)})_{\mathcal{G}(k,i)} = (\hat{\mathbf{w}}^{(k)})_i$.

Solution via ADMM. We now formulate a solution that includes coupling between the clusters applying ADMM, to produce a better estimate of the shared weights. In the workflow of ADMM, the objective should be transformed to

$$\text{minimize } \Phi(\mathbf{x}) + \Psi(\mathbf{z}) \quad \text{s.t. } \mathbf{G}\mathbf{x} + \mathbf{F}\mathbf{z} = \mathbf{c}, \tag{3}$$

by choosing $\Phi(\cdot)$, $\Psi(\cdot)$ and the constraints. We dualize on the regularization term, i.e., $\Psi(\mathbf{z}) = \alpha \mathbf{1}^T \mathbf{z}$, $\Phi(\mathbf{x}) = \| f(\mathbf{x}) - \widehat{\mathbf{b}} \|_2^2$, $\mathbf{G} = \mathbf{I}$, $\mathbf{F} = -\mathbf{I}$, $\mathbf{c} = \mathbf{0}$, getting the general form consensus with regularization [Boyd et al. 2011], with the following ADMM updates at each iteration $t + 1$:

$$\begin{aligned} \mathbf{x}_{t+1}^{(k)} &\in \underset{\mathbf{0} \leq \mathbf{x} \leq \mathbf{1}}{\text{argmin}} \left(\| f^{(k)}(\mathbf{x}) - \widehat{\mathbf{b}}^{(k)} \|_2^2 + \rho \| \mathbf{x} - \tilde{\mathbf{z}}_t^{(k)} + \mathbf{u}_t^{(k)} \|_2^2 \right), \\ \mathbf{z}_{t+1} &= \mathbf{S}^{-1} \left(\sum_{k=1}^{K} \mathbf{q}^{(k)} - \frac{\alpha_k}{\rho} \right), \qquad \mathbf{u}_{t+1}^{(k)} = \mathbf{u}_t^{(k)} + \mathbf{x}_{t+1}^{(k)} - \mathbf{z}_{t+1}^{(k)}. \end{aligned} \tag{4}$$

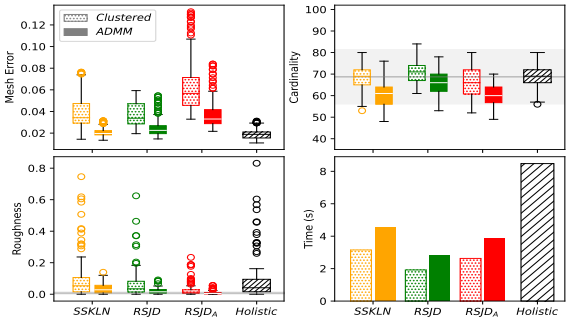

Figure 5: Results over the test set. Dotted-face bars represent naive clustered solutions, while the same-color solid bars are the corresponding ADMM solutions. A gray horizontal line shows the metric value of the ground-truth data.

A vector $\tilde{\mathbf{z}}^{(k)} \in \mathbb{R}^{m_k}$ is a local copy of the global variable $\mathbf{z} \in \mathbb{R}^m$, i.e., $(\tilde{\mathbf{z}}^{(k)})_i = \mathbf{z}_{\mathcal{G}(k,i)}$. The entries of $\mathbf{q}^{(k)} \in \mathbb{R}^m$ are the values of $\mathbf{x}_{t+1}^{(k)} + \mathbf{u}_t^{(k)}$ obtained for the corresponding cluster, i.e., $(\mathbf{q}^{(k)})_{\mathcal{G}(k,i)} = (\mathbf{x}_{t+1}^{(k)})_i + (\mathbf{u}_t^{(k)})_i$. Further, we proceed by solving the x-update step following the approach of [Racković et al. 2023a].

3 RESULTS

For each clustering strategy, we observe two approaches: I) a naive clustered solution, where (2) is solved independently for each cluster, and shared weights are averaged in the end; II) the proposed ADMM approach (4), where the shared components are constrained to be similar, by coupling between the local and global variables, to get more confident estimates. Additionally, we include a holistic case, i.e., the method of [Racković et al. 2023a], where problem (1) is solved without segmentation.

Mesh error is computed as a root mean squared error (RMSE) between the target mesh $\widehat{\mathbf{b}}$ and the estimated $f(\hat{\mathbf{w}})$, where $\hat{\mathbf{w}}$ is the estimated weight vector. Cardinality is the number of non-zero weights in $\hat{\mathbf{w}}$. The character *Jesse* (©unrealengine.com), is animated to give a wide range of motion. We need to choose a good value of α, for each approach, and the optimal K for *RSJD* and *RSJD*$_{\text{A}}$. We run experiments on 300 training frames with varying $\alpha > 0$ and $0 \leq K \leq m = 102$ (Fig. 3), and choose α at which cardinality is equal to ground-truth, as in this sense we have a fair comparison of different methods. The results of the *Sparse* approach are extremely poor, hence, we will dismiss it from further consideration. In all the other cases, the results obtained using ADMM outperform those of a naive clustered solution. For *RSJD* and *RSJD*$_{\text{A}}$, we need an optimal K, based primarily on the error vs cardinality trade-off, but also the execution time. For both methods, $K = 4$ leads to the best ADMM trade-off, yet, the time is considerably longer than for other choices of K. For *RSJD* we choose $K = 22$, as it gives only a slightly worse curve in the case of ADMM, while the time is almost half of the case with $K = 4$, and the results of a simple clustered method are the best performing for this choice. For *RSJD*$_{\text{A}}$, an increase in K leads to a decrease in the overall trade-off, and with $K = 20$ the execution time is as low as it gets.

The selected results are presented in Fig. 4. ADMM significantly improves the results compared to the naive clustered approach. The trade-off curve of *SSKLN* (using ADMM) is the only one reaching the accuracy of a holistic model, yet its execution time is the largest of the three distributed methods. The trade-off of E_R versus E_D and E_{ID} confirms the cross-validation would lead to choosing *RSJD* clustering with smaller E_R, and *RSJD*$_{\text{A}}$ clustering with smaller E_D. Also, we can see a relationship between the execution time and E_D, as the denser clusterings lead to a longer execution.

Test results in Fig. 5 show a clear distinction — in all three cases, the upper quartile of the ADMM solution is lower than the lower quartile of clustered solution. ADMM under *SSKLN* is comparable to the holistic in terms of median and quartiles, while with *RSJD* it is just slightly worse. The execution time of the clustered solution is lower than that of ADMM, yet the difference is not as large as between the holistic case to others. Finally, since the test set consists of an animation sequence, we are interested in the temporal smoothness, expressed using the *Roughness* penalty (lower values of correspond to smoother curves). The values for ADMM are significantly lower than the corresponding values for a naive clustered approach or holistic case, as can be also noticed in the smooth animation in supplementary video.

The application of ADMM on the clustered face leads to significant outperformance compared to the previous approaches that solved each cluster independently and averaged the shared components. ADMM produces visibly lower values of each metric, except of the execution time. However, the execution time under ADMM is still less than half of the holistic approach.

ACKNOWLEDGMENTS

This work has received funding from the Marie Skłodowska-Curie grant (812912), NOVA LINCS (FCT UIDB/ 04516/ 2020), DSAIPA/AI/ 0087/2018, and the Ministry of Education of the Republic of Serbia (451-03-9/2021-14/200125).

REFERENCES

Stephen Boyd, Neal Parikh, Eric Chu, Borja Peleato, Jonathan Eckstein, et al. 2011. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends® in Machine learning* 3, 1 (2011), 1–122.

Cumhur Ozan Çetinaslan. 2016. *Position Manipulation Techniques for Facial Animation.* Ph. D. Dissertation. Faculdade de Ciencias da Universidade do Porto.

Byoungwon Choe and Hyeong-Seok Ko. 2006. Analysis and synthesis of facial expressions with hand-generated muscle actuation basis. In *ACM SIGGRAPH 2006.*

John P. Lewis, Ken Anjyo, Taehyun Rhee, Mengjie Zhang, Frederic H. Pighin, and Zhigang Deng. 2014. Practice and Theory of Blendshape Facial Models. *Eurographics (State of the Art Reports)* 1, 8 (2014), 2.

Stevo Racković, Cláudia Soares, Dušan Jakovetić, and Zoranka Desnica. 2023a. High-fidelity Interpretable Inverse Rig: An Accurate and Sparse Solution Optimizing the Quartic Blendshape Model. *arXiv preprint arXiv:2302.04820* (2023).

Stevo Racković, Cláudia Soares, Dušan Jakovetić, and Zoranka Desnica. 2023b. A majorization–minimization-based method for nonconvex inverse rig problems in facial animation: algorithm derivation. *Optimization Letters* (2023), 1–15.

Stevo Racković, Cláudia Soares, Dušan Jakovetić, Zoranka Desnica, and Relja Ljubobratović. 2021. Clustering of the Blendshape Facial Model. In *2021 29th European Signal Processing Conference (EUSIPCO).* 1556–1560.

Yeongho Seol, Jaewoo Seo, Paul Hyunjin Kim, J. P. Lewis, and Junyong Noh. 2011. Artist Friendly Facial Animation Retargeting. In *SIGGRAPH Asia 2011.*

J. Rafael Tena, Fernando De la Torre, and Iain Matthews. 2011. Interactive Region-Based Linear 3D Face Models. In *ACM SIGGRAPH 2011 Papers.*